Hotel Database Application Project

Zhi Zhang, Shiqi Tu, Anqi Chen

Project Phase 1

Description of the Database Application

For our team project, we plan to develop a hotel management database application. This database is designed to help the 465 Hotel manage its day-to-day operations efficiently and provide exceptional experiences for its guests. The database will store information about the hotel's details, room and its types, bookings, and services.

The application will also enable the hotel to keep track of its employees, their departments, as well as the personal details of its guests to provide better service. With this database application, the 465 Hotel will be able to optimize its management systems, attract more customers during the 2024 Paris Olympics, and provide an superior experience for all its guests.

Case

The 465 Hotel is a luxury hotel situated in the heart of Paris, and with the upcoming 2024 Paris Olympics, the hotel management is looking to optimize its systems to attract more guests during the games.

To give guests a more details understanding of the hotel, its hotel description is very comprehensive, which includes hotel ID, hotel name, contact info(number and email), address, and rating. The hotel contains 465 rooms, and each room is identified by its ID, and is described by the room size and capacity.

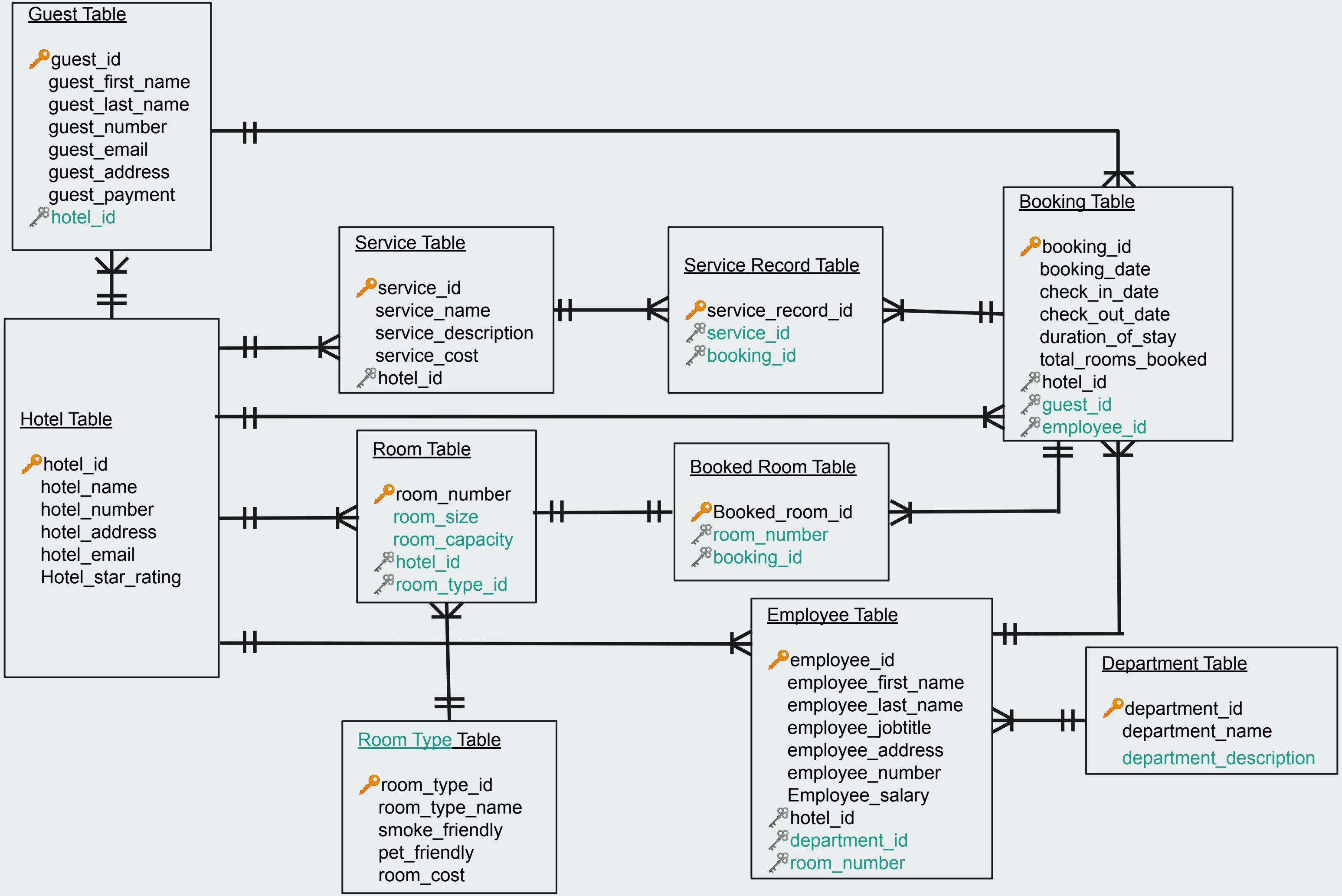
The hotel offers different types of rooms, including standard, deluxe, and suite, with each room type containing the room name, number, cost, and description (smoke and pet policy). When the guests make bookings, the hotel wants to record details such as the guest ID, booking date, check-in date, check-out date, duration of stay, and the number of rooms booked. Guests can make bookings for multiple rooms, and each booking can only be made by one guest.

Case

The hotel also provides various services that guests can use during their stay. Each service is uniquely identified by services ID, names, costs, and description. Guests can use multiple hotel services during their stay.

The 465 Hotel employs a team of employees in different departments, including housekeeping, front desk, restaurant, and maintenance. Each employee has a unique employee ID, and is also characterized by their name, job title, contact info (address and number), department ID, and salary. For each department, it has a unique department ID, and is also described by department name and description. Each employee can only work in one department, and each department at least has one employee.

The hotel also wants to keep track of the personal details of the hotel's guests, for each guests the hotel need to store guest ID, name, phone number, email, address, and preferred payment method. Thereby, they can send out some promotions or updates of the hotels to provide a better service.



Guest Table

guest_id
guest_first_name
guest_last_name
guest_number
guest_email
guest_address
guest_payment
hotel_id

Hotel Table

hotel_id
hotel_name
hotel_number
hotel_address
hotel_email
Hotel_star_rating

Service Table

service_id
service_name
service_description
service_cost
hotel_id

Service Record Table

service_record_id
service_id
booking_id

Booking Table

booking_id
booking_date
check_in_date
check_out_date
duration_of_stay
total_rooms_booked
hotel_id
guest_id
employee_id

Room Table

room_number
room_size
room_capacity
hotel_id
room_type_id

Booked Room Table

Booked_room_id
room_number
booking_id

Room Type Table

room_type_id
room_type_name
smoke_friendly
pet_friendly
room_cost

Employee Table

employee_id
employee_first_name
employee_last_name
employee_jobtitle
employee_address
employee_number
Employee_salary
hotel_id
department_id
room_number

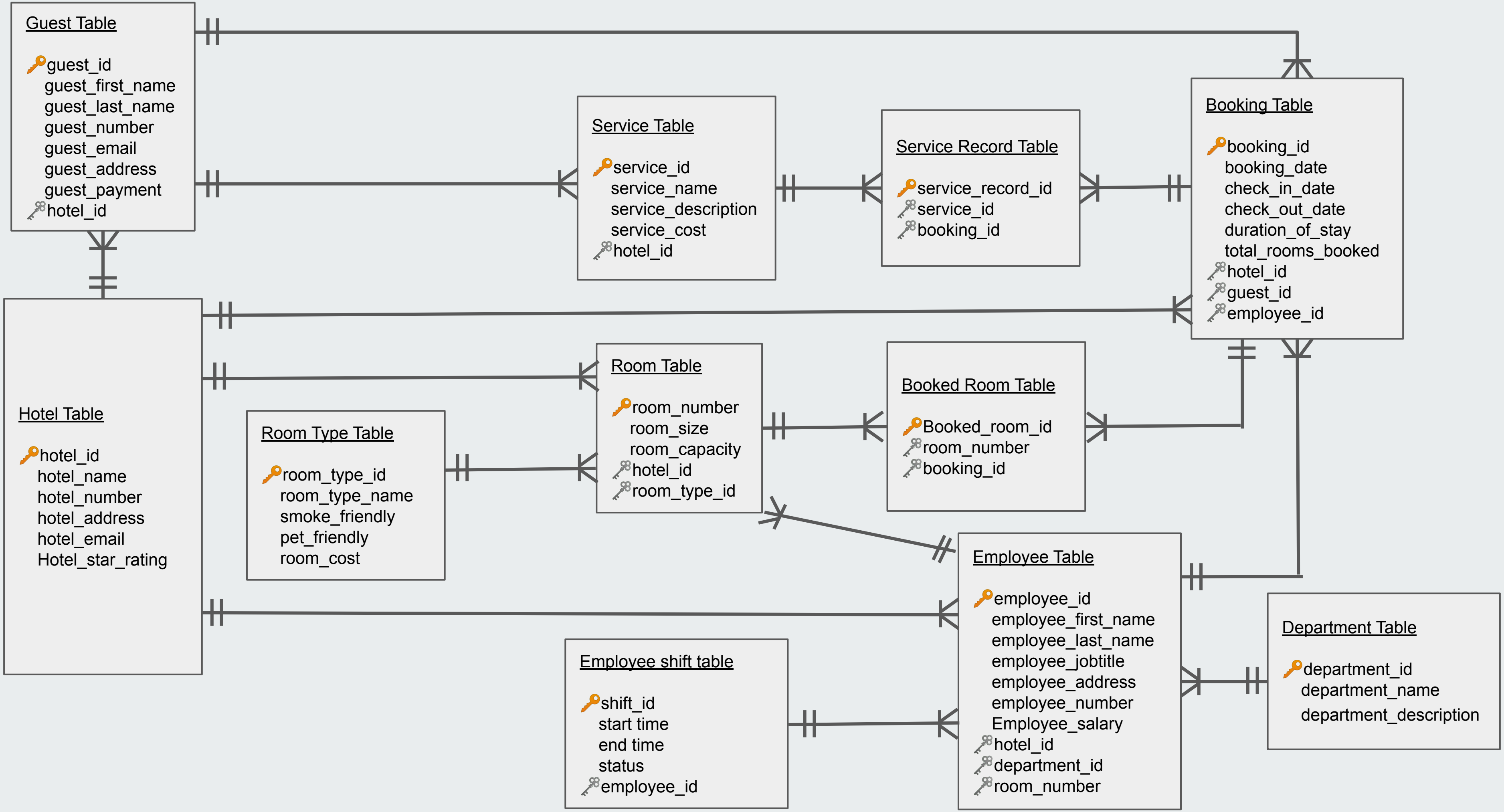
Department Table

department_id
department_name
department_description

Instances of Each Entity

- In guest table, one instance would be guest_id = 00000001, guest_first_name = William, guest_last name = Howard, guest_number = 1, guest_email = williamhoward@gmail.com, guest_address = 175 freeman st, guest_payment=credit card, hotel_id = 465
- In hotel table, one instance would be hotel_id = 456, hotel_name = The 465 hotel, hotel_number = 001100, hotel_address = 595 commonwealth ave, hotel_email = the465hotel@gmail.com, hotel_star_rating = 5
- In booking table, one instance would be booking_id = 11223344, booking_date = 03/02/2023, check_in_date = 03/05/2023, check_out_date = 03/07/2023, duration_of_stay = 2, total_rooms_booked = 1, hotel_id = 465, guest_id = 00000001, employee_id = 00001
- In service table, one instance would be service_id = 171717, service_name = Food & Beverage, service_description = 2 package of chips, service_cost = $10, hotel_id = 456
- In service record table, one instance would be service_record_id = 111111, service_id = 171717, and booking_id = 11223344
- In room type table, one instance would be room_type_id = 0001, room_type_name = single, smoke_friendly = no, pet_friendly = yes, room_cost = $300
- In room table, one instance would be room_number = 1224, room_size = 300, hotel_id = 456, room_type_id = 0001
- In booked room table, one instance would be booked_room_id = 112233445, room_number = 1224, booking_id = 11223344
- In department table, one instance would be department_id = 4321, department_name = front office
- In employee table, one instance would be employee_id = 00634, employee_first_name = Emily, employee_last_name = Turner, employee_jobtitle = Manager, employee_address = 1505 commonwealth avenue, employee_number = 8577831657, employee_salary = $75000, employee_shift = day, hotel_id = 456, department_id = Manager, room_number = null
- In employee shift table, one instance would be shift id = 0012345, start time = 9am, end time = 5 pm, status = completed, employee_id = 00383

Project Phase 2



Guest Table

guest_id
guest_first_name
guest_last_name
guest_number
guest_email
guest_address
guest_payment
hotel_id

Service Table

service_id
service_name
service_description
service_cost
hotel_id

Service Record Table

service_record_id
service_id
booking_id

Booking Table

booking_id
booking_date
check_in_date
check_out_date
duration_of_stay
total_rooms_booked
hotel_id
guest_id
employee_id

Hotel Table

hotel_id
hotel_name
hotel_number
hotel_address
hotel_email
Hotel_star_rating

Room Table

room_number
room_size
room_capacity
hotel_id
room_type_id

Booked Room Table

Booked_room_id
room_number
booking_id

Room Type Table

room_type_id
room_type_name
smoke_friendly
pet_friendly
room_cost

Employee Table

employee_id
employee_first_name
employee_last_name
employee_jobtitle
employee_address
employee_number
Employee_salary
hotel_id
department_id
room_number

Department Table

department_id
department_name
department_description

Employee shift table

shift_id
start time
end time
status
employee_id

Relational schema (bold is primary key, underline is foreign key)

Hotel_table (**hotel_id**, hotel_name, hotel_number, hotel_address, hotel_email, Hotel_star_rating)

Guest_table (**guest_id**, guest_first_name, guest_last_name, guest_number, guest_email, guest_address, guest_payment, hotel_id)

Service_table (**service_id**, service_name, service_description, service_cost, hotel_id)

Department _table (**department_id**, department_name, department_description)

Employee _table (**employee_id**, employee_first_name, employee_last_name, employee_jobtitle, employee_address, employee_number, employee_salary, employee_shift, hotel_id, department_id, room_number)

Booking_table (booking_id, booking_date, check_in_date, check_out_date, duration_of_stay, total_rooms_booked, hotel_id, guest_id, employee_id)

Service_Record_table (**service_record_id,** service_id, booking_id)

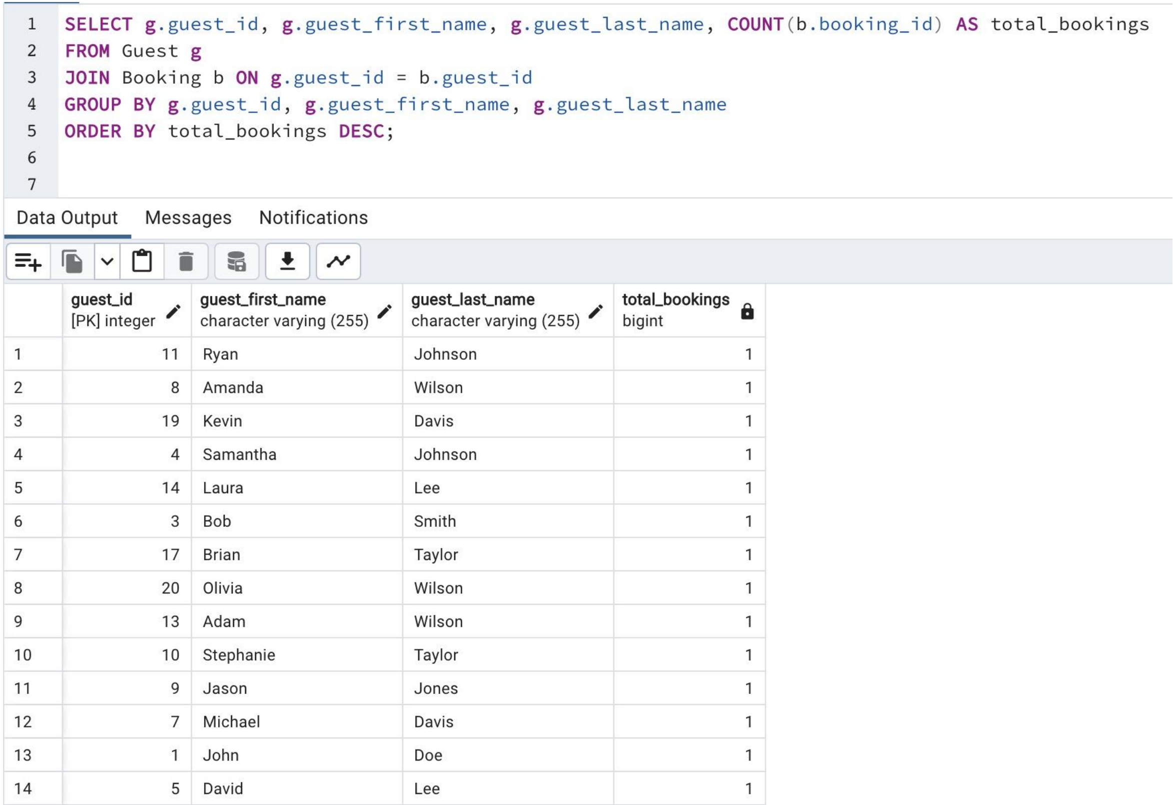
Room_Type_table (**room_type_id**, room_type_name, smoke_friendly, pet_friendly, room_cost)

Room_table (**room_number**, room_size, room_capacity, hotel_id, room_type_id)

Booked_Room_table (**Booked_room_id**, room_number, booking_id)

Employee_Shift_table (**shift_id**, start time, end time, status, employee_id)

Query #1 Retrieve the total number of bookings made by each guest, and sort them in descending order

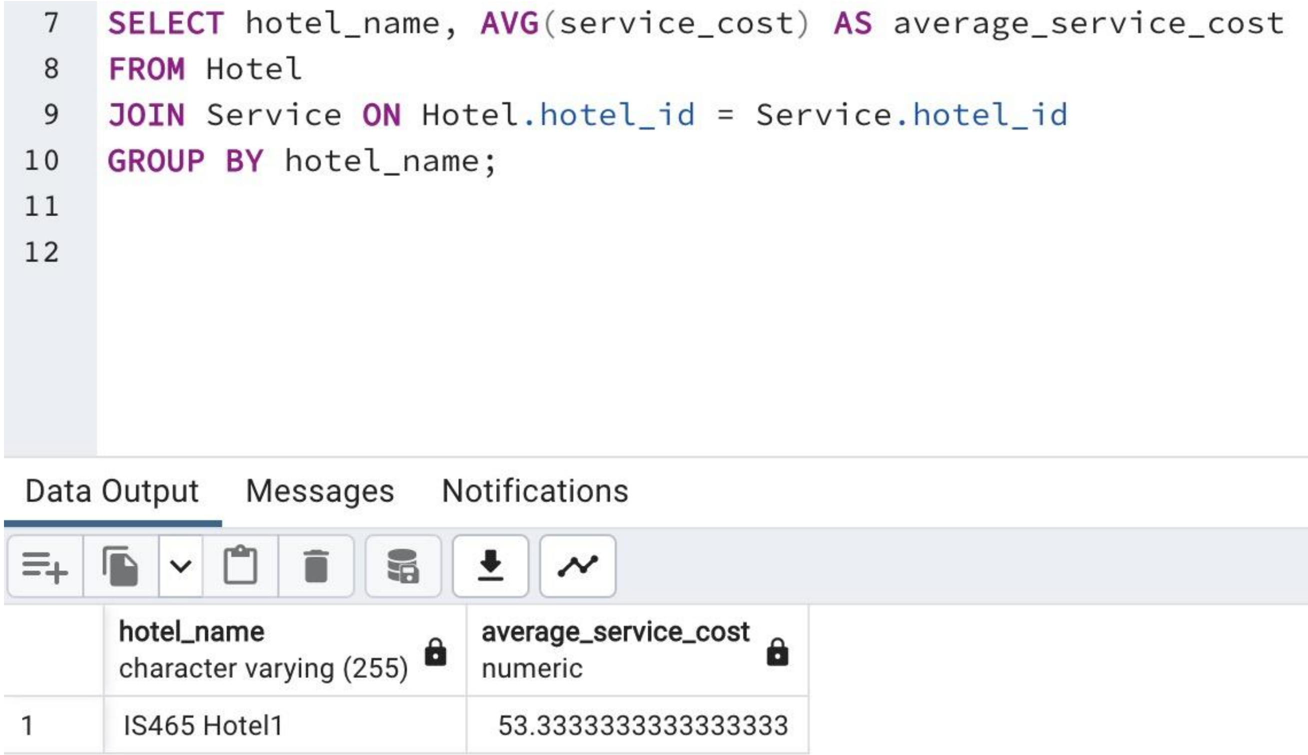


```sql
SELECT g.guest_id, g.guest_first_name, g.guest_last_name, COUNT(b.booking_id) AS total_bookings
FROM Guest g
JOIN Booking b ON g.guest_id = b.guest_id
GROUP BY g.guest_id, g.guest_first_name, g.guest_last_name
ORDER BY total_bookings DESC;
```

	guest_id [PK] integer	guest_first_name character varying (255)	guest_last_name character varying (255)	total_bookings bigint
1	11	Ryan	Johnson	1
2	8	Amanda	Wilson	1
3	19	Kevin	Davis	1
4	4	Samantha	Johnson	1
5	14	Laura	Lee	1
6	3	Bob	Smith	1
7	17	Brian	Taylor	1
8	20	Olivia	Wilson	1
9	13	Adam	Wilson	1
10	10	Stephanie	Taylor	1
11	9	Jason	Jones	1
12	7	Michael	Davis	1
13	1	John	Doe	1
14	5	David	Lee	1

This query can help the manager to identify loyal customers and provide special offers or promotions to retain them. By sorting the results in descending order, the manager can also identify guests who have only made a single booking and devise strategies to encourage them to return.

Query #2 Retrieve the average cost of services provided by each hotel

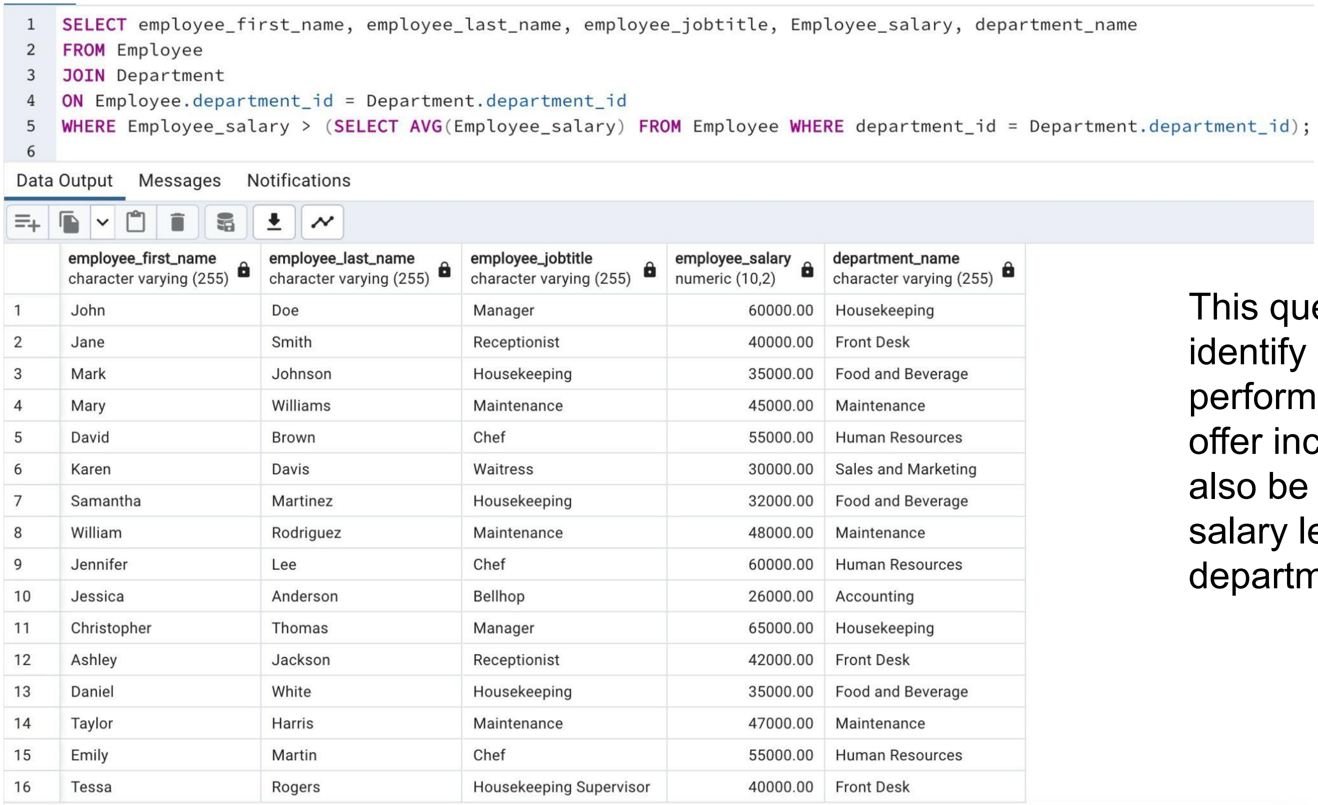


```sql
SELECT hotel_name, AVG(service_cost) AS average_service_cost
FROM Hotel
JOIN Service ON Hotel.hotel_id = Service.hotel_id
GROUP BY hotel_name;
```

	hotel_name character varying (255)	average_service_cost numeric
1	IS465 Hotel1	53.3333333333333333

This query can help the manager to identify the most profitable services and improve the hotel's revenue. The manager can also compare the average service cost across different hotels and identify opportunities to optimize costs or increase prices.

Query #3 Retrieve the list of employees who earn more than the average sa of their department



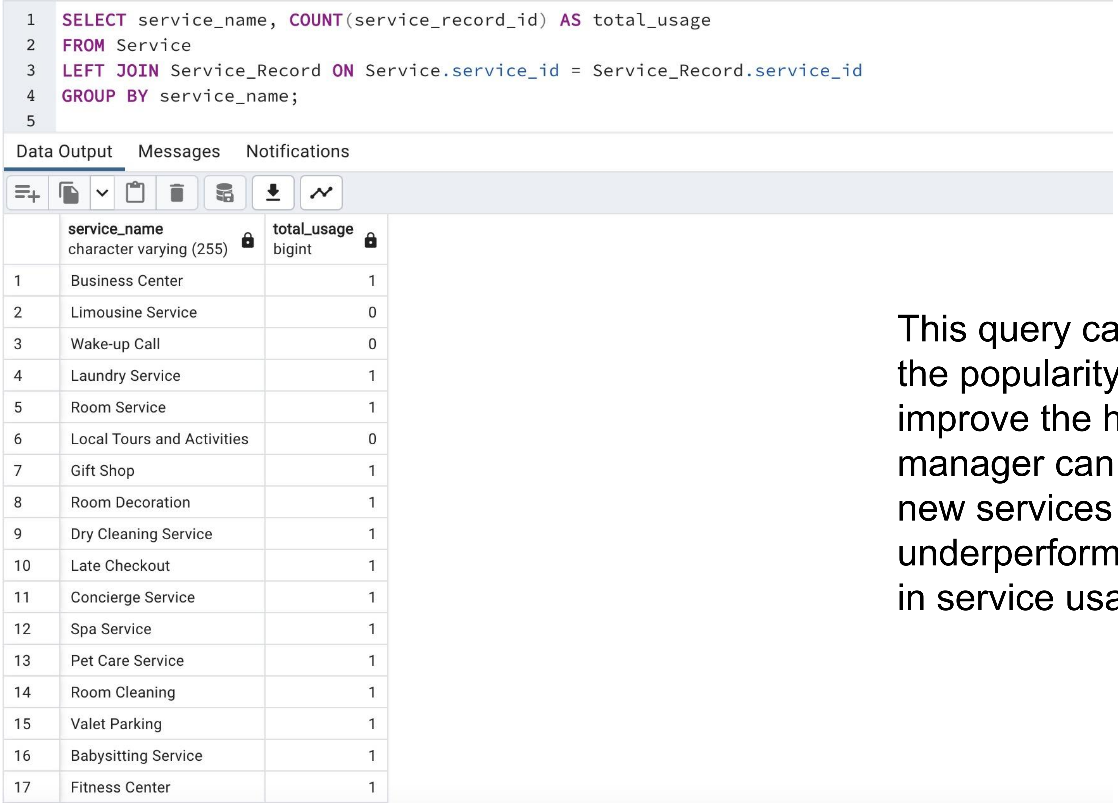
```sql
SELECT employee_first_name, employee_last_name, employee_jobtitle, Employee_salary, department_name
FROM Employee
JOIN Department
ON Employee.department_id = Department.department_id
WHERE Employee_salary > (SELECT AVG(Employee_salary) FROM Employee WHERE department_id = Department.department_id);

```

	employee_first_name character varying (255)	employee_last_name character varying (255)	employee_jobtitle character varying (255)	employee_salary numeric (10,2)	department_name character varying (255)
1	John	Doe	Manager	60000.00	Housekeeping
2	Jane	Smith	Receptionist	40000.00	Front Desk
3	Mark	Johnson	Housekeeping	35000.00	Food and Beverage
4	Mary	Williams	Maintenance	45000.00	Maintenance
5	David	Brown	Chef	55000.00	Human Resources
6	Karen	Davis	Waitress	30000.00	Sales and Marketing
7	Samantha	Martinez	Housekeeping	32000.00	Food and Beverage
8	William	Rodriguez	Maintenance	48000.00	Maintenance
9	Jennifer	Lee	Chef	60000.00	Human Resources
10	Jessica	Anderson	Bellhop	26000.00	Accounting
11	Christopher	Thomas	Manager	65000.00	Housekeeping
12	Ashley	Jackson	Receptionist	42000.00	Front Desk
13	Daniel	White	Housekeeping	35000.00	Food and Beverage
14	Taylor	Harris	Maintenance	47000.00	Maintenance
15	Emily	Martin	Chef	55000.00	Human Resources
16	Tessa	Rogers	Housekeeping Supervisor	40000.00	Front Desk

This query can help the manager to identify employees who are performing better than average and offer incentives to retain them. It can also be used to find any differences in salary levels across various departments.

Query #4 Retrieve the list of all services provided by the hotel along with the number of times each service has been used.

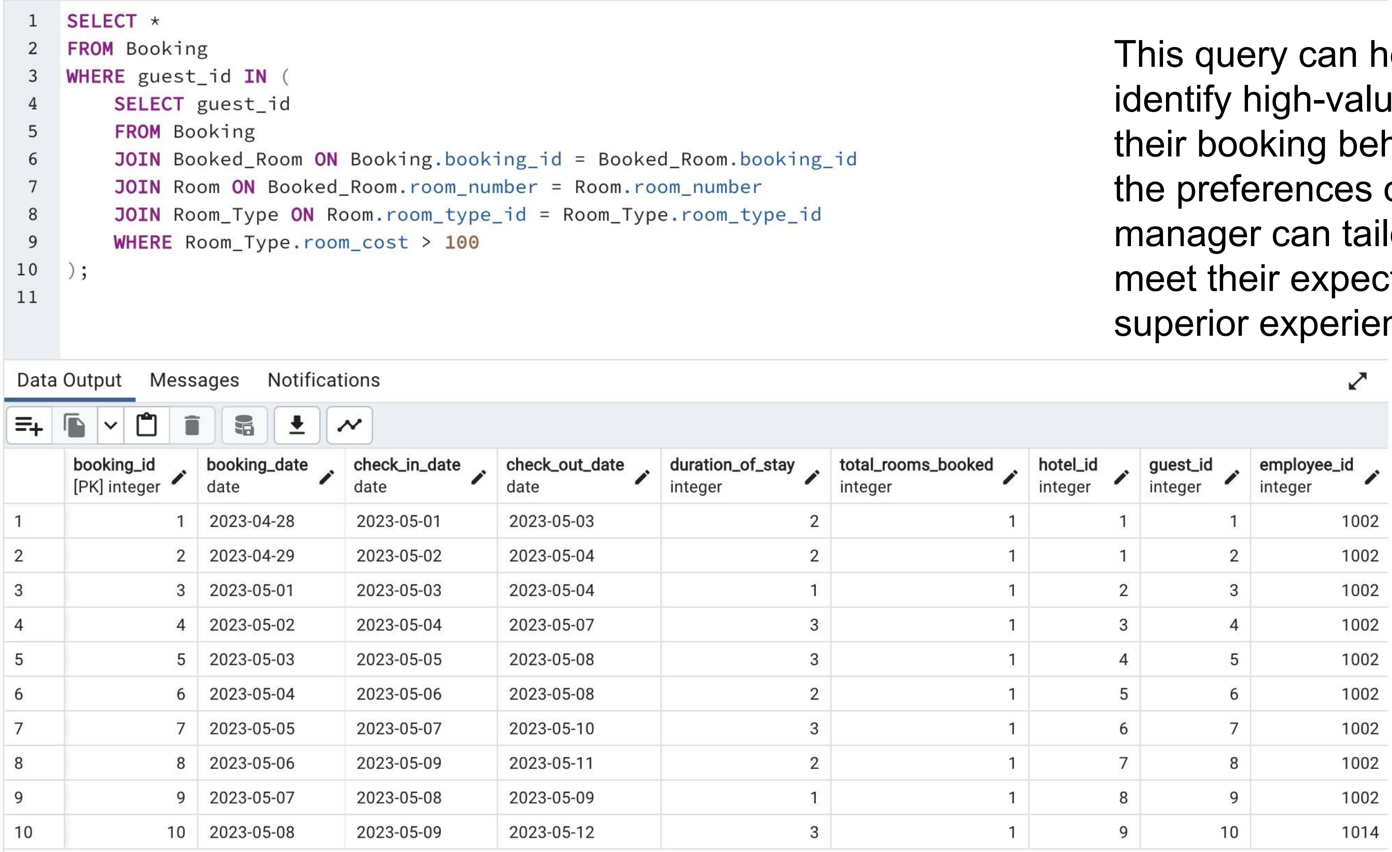


```sql
SELECT service_name, COUNT(service_record_id) AS total_usage
FROM Service
LEFT JOIN Service_Record ON Service.service_id = Service_Record.service_id
GROUP BY service_name;
```

	service_name character varying (255)	total_usage bigint
1	Business Center	1
2	Limousine Service	0
3	Wake-up Call	0
4	Laundry Service	1
5	Room Service	1
6	Local Tours and Activities	0
7	Gift Shop	1
8	Room Decoration	1
9	Dry Cleaning Service	1
10	Late Checkout	1
11	Concierge Service	1
12	Spa Service	1
13	Pet Care Service	1
14	Room Cleaning	1
15	Valet Parking	1
16	Babysitting Service	1
17	Fitness Center	1

This query can help the manager to analyze the popularity of various services and improve the hotel's service options. The manager can find opportunities to launch new services or discontinue underperforming ones by examining trends in service usage.

Query #5 Retrieve the list of all bookings made by guests who have stayed in rooms costing more than 100 euros per night.



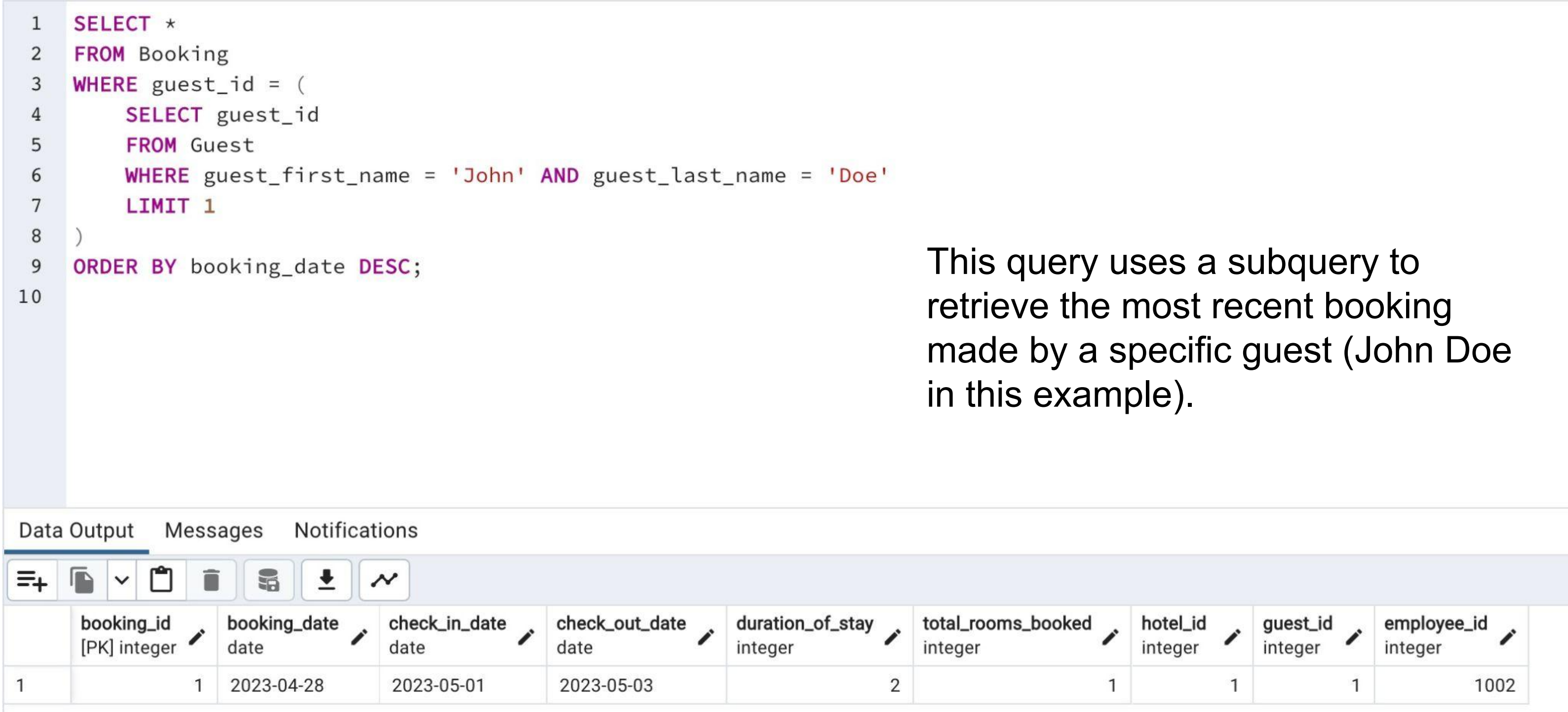
```sql
SELECT *
FROM Booking
WHERE guest_id IN (
    SELECT guest_id
    FROM Booking
    JOIN Booked_Room ON Booking.booking_id = Booked_Room.booking_id
    JOIN Room ON Booked_Room.room_number = Room.room_number
    JOIN Room_Type ON Room.room_type_id = Room_Type.room_type_id
    WHERE Room_Type.room_cost > 100
);

```

	booking_id [PK] integer	booking_date date	check_in_date date	check_out_date date	duration_of_stay integer	total_rooms_booked integer	hotel_id integer	guest_id integer	employee_id integer
1	1	2023-04-28	2023-05-01	2023-05-03	2	1	1	1	1002
2	2	2023-04-29	2023-05-02	2023-05-04	2	1	1	2	1002
3	3	2023-05-01	2023-05-03	2023-05-04	1	1	2	3	1002
4	4	2023-05-02	2023-05-04	2023-05-07	3	1	3	4	1002
5	5	2023-05-03	2023-05-05	2023-05-08	3	1	4	5	1002
6	6	2023-05-04	2023-05-06	2023-05-08	2	1	5	6	1002
7	7	2023-05-05	2023-05-07	2023-05-10	3	1	6	7	1002
8	8	2023-05-06	2023-05-09	2023-05-11	2	1	7	8	1002
9	9	2023-05-07	2023-05-08	2023-05-09	1	1	8	9	1002
10	10	2023-05-08	2023-05-09	2023-05-12	3	1	9	10	1014

This query can help the manager to identify high-value guests and analyze their booking behavior. By understanding the preferences of these guests, the manager can tailor the hotel's services to meet their expectations and provide a superior experience.

Query #6 Getting details of the most recent booking made by a specific guest



```sql
SELECT *
FROM Booking
WHERE guest_id = (
    SELECT guest_id
    FROM Guest
    WHERE guest_first_name = 'John' AND guest_last_name = 'Doe'
    LIMIT 1
)
ORDER BY booking_date DESC;

```

	booking_id [PK] integer	booking_date date	check_in_date date	check_out_date date	duration_of_stay integer	total_rooms_booked integer	hotel_id integer	guest_id integer	employee_id integer
1	1	2023-04-28	2023-05-01	2023-05-03	2	1	1	1	1002

This query uses a subquery to retrieve the most recent booking made by a specific guest (John Doe in this example).

This query will be useful for the hotel manager to provide personalized service to repeat guests and identify any issues or complaints related to their most recent stay.